```
 1 import cv2
 2 from cvzone.HandTrackingModule import HandDetector
 3 from cvzone.ClassificationModule import Classifier
 4 import numpy as np
 5 import math
 6
 7 cap = cv2.VideoCapture(0)
 8 detector = HandDetector(maxHands=1)
 9 classifier = Classifier("Model/keras_model.h5","Model
   /labels.txt")
10
11 offset = 20
12 imgSize = 300
13
14 folder = "Data/Z"
15 counter = 0
16
17 labels = ["A", "B", "C", "D", "E", "F", "G", "H", "I"
   , "J", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "
   T", "U", "V", "W", "X", "Y", "Z"]
18
19 while True:
20     success, img = cap.read()
21     imgOutput = img.copy()
22     hands, img = detector.findHands(img)
23     if hands:
24         hand = hands[0]
25         x, y, w, h = hand['bbox']
26
27         imgWhite = np.ones((imgSize, imgSize, 3), np.
   uint8)*255
28         imgCrop = img[y - offset : y + h + offset , x
    - offset : x + w + offset]
29
30         imgCropShape = imgCrop.shape
31
32         aspectRatio = h/w
33
34         if aspectRatio > 1:
35             k = imgSize / h
36             wCal = math.ceil(k * w)
```

```
37              imgResize = cv2.resize(imgCrop,(wCal,
   imgSize))
38              imgResizeShape = imgResize.shape
39              wGap = math.ceil((imgSize-wCal)/2)
40              imgWhite[:, wGap:wCal + wGap] = imgResize
41              prediction, index = classifier.
   getPrediction(imgWhite, draw = False)
42              print(prediction, index)
43
44          else:
45              k = imgSize / w
46              hCal = math.ceil(k * h)
47              imgResize = cv2.resize(imgCrop, (imgSize
   , hCal))
48              imgResizeShape = imgResize.shape
49              hGap = math.ceil((imgSize - hCal) / 2)
50              imgWhite[hGap:hCal + hGap, :] = imgResize
51              prediction, index = classifier.
   getPrediction(imgWhite, draw = False)
52
53          cv2.rectangle(imgOutput, (x - offset, y -
   offset - 50), (x - offset + 90 , y - offset - 50 + 50
   ), (255, 0, 255), cv2.FILLED)
54          cv2.putText(imgOutput, labels[index], (x, y-
   26), cv2.FONT_HERSHEY_COMPLEX, 1.7, (255, 255, 255),
   2)
55          cv2.rectangle(imgOutput, (x-offset,y-offset
   ), (x+w+offset, y+h+offset), (255, 0, 255), 4)
56
57          cv2.imshow("ImageCrop", imgCrop)
58          cv2.imshow("ImageWhite", imgWhite)
59
60      cv2.imshow("Image", imgOutput)
61      cv2.waitKey(1)
62
```